

14



El impacto de BASIC en la transformación digital del siglo XX: de la democratización de la alfabetización informática a la hegemonía del software

The impact of BASIC on the digital transformation of the 20th century: from the democratization of computer

**Jacobo Roda-Segarra;
Ana Martí-García;
Santiago Mengual-Andrés;
Andrés Payà Rico***

DOI: 10.5944/reec.48.2025.45282

**Recibido: 6 de mayo de 2025
Aceptado: 17 de julio de 2025**

*JACOBO RODA-SEGARRA; ANA MARTÍ-GARCÍA; SANTIAGO MENGUAL-ANDRÉS; ANDRÉS PAYÀ RICO: Universitat de València. **Datos de contacto:** Email: jacobo.roda@uv.es; Ana.Marti-Garcia@uv.es; santiago.mengual@uv.es; andres.paya@uv.es

Resumen

El lenguaje de programación BASIC, desarrollado en 1975 por Bill Gates y Paul Allen, estuvo presente en gran parte de los ordenadores más relevantes de la década de 1975-1985, simplificando la programación de un ordenador. Las prácticas en el marco de la educación informal tuvieron un papel muy relevante en el aprendizaje y difusión de BASIC. Se ha empleado un análisis de contenido de tipo mixto (cuantitativo y cualitativo), desde un enfoque longitudinal acotado y temático, de los contenidos relacionados con BASIC en los primeros 16 números de la revista estadounidense *Byte* (1975-1976), pionera en el ámbito de la divulgación informática doméstica. Se han analizado 231 contenidos entre artículos, editoriales, publicidad de ordenadores, reseñas o anuncios de libros, anuncios de clubes y cartas de lectores/as. Más de la mitad de los contenidos donde aparecía el término BASIC fueron anuncios de ordenadores. Los artículos, mucho más escasos, tuvieron tres fases o períodos diferenciados de publicación: en una primera más divulgativa se trataba de responder a la pregunta de la utilidad de un ordenador y, por extensión, de programar en BASIC; en una segunda los artículos se centraron en la realización práctica de programas; en la tercera irrumpieron los videojuegos, y la programación en BASIC cobró un nuevo sentido. La decisión de Microsoft de apostar por la democratización de la informática no fue la más lógica desde el punto de vista empresarial, pero sirvió para colocar a BASIC (y a Microsoft) en una posición estratégica en la incipiente industria de la informática doméstica. Con esta decisión se observan los primeros indicios de un filantropocapitalismo digital, enmascarado bajo un mensaje de democratización de la informática, pero con el objetivo de lograr una gran capacidad de influencia.

Palabras clave: BASIC; educación; TIC; aprendizaje informal; programación

Abstract

The BASIC programming language, developed in 1975 by Bill Gates and Paul Allen, was present in many of the most influential computers of the 1975–1985 decade, significantly simplifying computer programming. Informal education practices played a key role in the learning and dissemination of BASIC. A mixed-methods content analysis (quantitative and qualitative) was conducted using a longitudinal, thematically bounded approach. The study focused on content related to BASIC in the first 16 issues (1975–1976) of *Byte*, a pioneering U.S. magazine dedicated to home computing. A total of 231 items were analyzed, including articles, editorials, computer advertisements, book reviews or announcements, club ads, and letters from readers. More than half of the contents referencing BASIC were computer advertisements. Articles, which were considerably fewer, followed three distinct phases: the first was more introductory and addressed the usefulness of owning a computer and, by extension, programming in BASIC; the second phase emphasized the practical development of programs; and in the third phase, video games emerged, giving BASIC programming a new meaning. Microsoft's decision to promote the democratization of computing was not the most logical from a business perspective, but it positioned BASIC (and Microsoft) strategically within the emerging home computing industry. This decision reveals early signs of a form of digital philanthrocapitalism—presented under the guise of democratizing computing, but ultimately aiming to secure substantial influence.

Keywords: BASIC; education; ICT; informal education; programming

1. Máquinas programables y lenguajes de programación

Los ordenadores se engloban dentro de un tipo de máquinas denominadas programables, cuya característica fundamental es que su comportamiento no siempre es el mismo, sino que depende de una parte modificable conocida como programa (Cerrada *et al.*, 2001). El programa se articula mediante algoritmos, que son una serie finita de reglas y operaciones que resuelven un problema específico (Knuth, 1997). Este enfoque se fundamenta en las aportaciones teóricas de Alan Turing en 1936, cuya máquina universal permitió modelizar cualquier tipo de cálculo y, al mismo tiempo, estableció un límite superior en la capacidad de cálculo de cualquier ordenador presente o futuro (Brookshear, 1993). Así pues, los programas y sus algoritmos constituyentes se convierten en el elemento nuclear de un ordenador, dándole la característica versatilidad de comportarse de diferentes formas en función de las necesidades de las personas.

Sin embargo, los programas que un ordenador puede ejecutar de una manera directa son una serie de instrucciones extremadamente complejas de escribir o interpretar por parte de una persona, ya que «normalmente contienen códigos numéricos sin ningún sentido nemotécnico, y compuestos por millares o incluso millones de operaciones elementales» (Cerrada *et al.*, 2001, p.10). Estos programas se denominan código máquina; su escritura e interpretación por parte de las personas se simplifica en cierta medida mediante el uso de ensambladores, que permiten el uso de un conjunto léxico reducido que evita el uso de los códigos numéricos del código máquina. A pesar de la ventaja que aportan los ensambladores, programar un ordenador mediante ensambladores conlleva dos problemas fundamentales: 1) a pesar de que el ensamblador humaniza en cierta medida el código máquina, sigue siendo un proceso muy complejo no accesible para todas las personas, y 2) cada tipo de ordenador tiene su propio registro de instrucciones, con lo que el código máquina producido para un ordenador no funcionará en otro diferente.

Una solución para esta problemática vino de la mano de los lenguajes de programación, que son sistemas formales de signos y reglas que especifican la lógica y el comportamiento del programa (Reynolds, 2008) mediante un lenguaje muy simplificado, pero que tiene ciertas similitudes con el lenguaje natural. En este sentido, McCracken *et al.* (1957) afirmaron de una manera incipiente que programar era el proceso de traducir desde el lenguaje conveniente para los seres humanos al lenguaje conveniente para los ordenadores. El código máquina y los ensambladores son considerados lenguajes de programación de bajo nivel, mientras que los lenguajes de programación que se acercan al lenguaje natural se consideran de alto nivel, y se diseñaron para mejorar la legibilidad y productividad de programadores y programadoras (Sebesta, 2007). Un ejemplo del contraste que existe entre la dificultad de escribir programas de bajo nivel frente a aquellos con lenguajes de alto nivel puede observarse en la Figura 1. Esta figura representa un mismo programa que escribe «HOLA» por pantalla en un ZX Spectrum, pero con tres lenguajes diferentes: código máquina, ensamblador y BASIC.

MICROPROCESADOR	BAJO NIVEL		ALTO NIVEL	USUARIO/A
	Código máquina	Ensamblador	BASIC	
	21 00 40 11 00 00 01 04 00 ED B0	LD HL, mensaje CALL print_string RET mensaje: DB "Hola",13,10,0 print_string: LD A, (HL) CP 0 RET Z CALL 0x10 INC HL JP print_string	PRINT "HOLA"	

Figura 1. Comparativa entre lenguajes de bajo y alto nivel. Fuente: elaboración propia.

La simplicidad del enfoque de los lenguajes de alto nivel solucionaba el primero de los problemas expuestos previamente, y es que algo extremadamente complejo como era programar un ordenador se convertía en una tarea mucho más sencilla y accesible, lo que abrió las puertas a la extensión de la enseñanza de esta disciplina a otros públicos. Adicionalmente, los lenguajes de alto nivel también solucionaban la segunda de las dificultades: si bien el código máquina se producía exclusivamente para un tipo de ordenador, el código escrito en un lenguaje de alto nivel podía ser ejecutado en una heterogeneidad de ordenadores, siempre que estos contaran con los intérpretes necesarios (Figura 2).

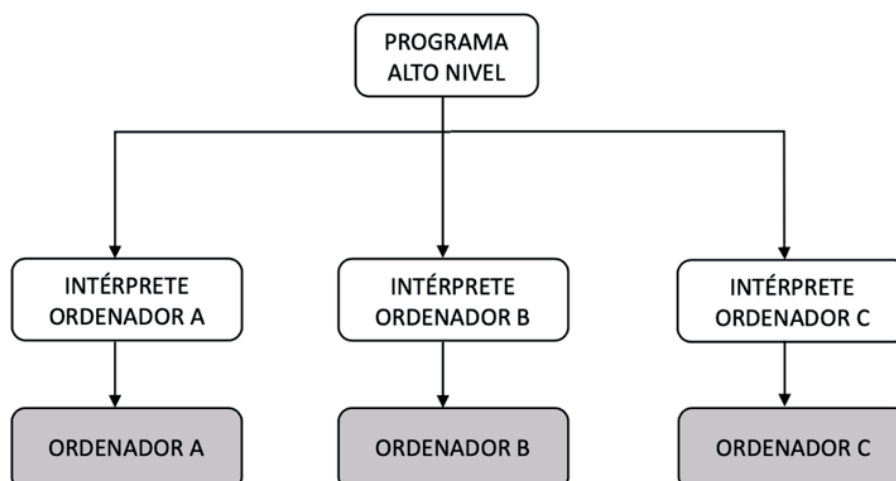


Figura 2. Esquema de funcionamiento de un programa de alto nivel en diferentes ordenadores. Fuente: elaboración propia.

Esta nueva necesidad de crear intérpretes para diferentes ordenadores suponía un esfuerzo importante para las compañías de desarrollo de software, pero para una recién nacida Microsoft fue una decisión estratégica con un impacto de universalización y extensión de este conocimiento enorme, tal como se mostrará en la presente investigación.

2.El nacimiento del lenguaje de alto nivel BASIC

En 1964, dos profesores del Dartmouth College de Hanover llamados John Kemeny y Thomas Kurtz desarrollaron un lenguaje de programación de alto nivel para su uso en la misma universidad. Su objetivo principal, tal como lo expresó Kurtz (1978), fue proveer a la comunidad de usuarios de un acceso «amigable» a los ordenadores. Fue denominado BASIC, acrónimo de Beginner's All-purpose Symbolic Instruction Code y, según The President's Science Advisory Committee de los Estados Unidos (1967), respondía a 5 criterios: facilidad de uso, fiabilidad, rapidez de respuesta, soporte satisfactorio y acceso gratuito; estas características, junto con una breve e intensa formación, preparaba a los miembros de la comunidad educativa del Dartmouth College a trabajar con un ordenador (Nevison, 1976).

Sus efectos fueron notables, ya que una década después (entre el verano de 1974 y la primavera de 1975) en un total de 184 asignaturas se recurrió al apoyo de los ordenadores, lo que supuso un 12 % de las asignaturas que se impartieron en el Dartmouth College durante ese intervalo; el potencial pedagógico de BASIC se extendió entre el personal docente de la universidad, ya que las asignaturas en las que se recurrió a los ordenadores no solo se ceñían a disciplinas técnicas, sino que también incluían muchas otras de los campos de las ciencias sociales y las humanidades, como antropología, arte, educación, geografía, historia, música, filosofía, psicología y sociología entre otras (Nevison, 1976).

En enero de 1975, la revista Popular Electronics publicó un número en el que, en la portada, se presentaba el primer ordenador personal, el Altair 8800 (Figura 3).



Figura 3. Anuncio del primer ordenador personal. Fuente: Portada de Popular Electronics, enero de 1975.

Tal como afirma Gates (2025) en su artículo celebrando el 50 aniversario del nacimiento de Microsoft, cuando Paul Allen y él leyeron el número de enero de 1975 de *Popular Electronics* se dieron cuenta de que la revolución de los ordenadores personales era inminente, y que ellos querían estar en «lo más alto». El Altair 8800, fabricado por la empresa MITS, inauguraba la era de los ordenadores personales de 8 bits, pero no disponía de software y llegar a ser un programador eficiente del ordenador requería «una buena cantidad de experiencia y una gran cantidad de creatividad»¹ (Roberts y Yates, 1975, p.34); el ordenador se programaba directamente en código binario mediante los conmutadores de la parte frontal, y los resultados se presentaban de la misma manera mediante indicadores luminosos.

Gates y Allen identificaron esta carencia de software pero, lejos de optar por desarrollar programas específicos para el Altair 8800 que resolvieran diferentes tareas, tomaron un rumbo diferente: decidieron decantarse por programar el intérprete de un lenguaje de programación. Tal como continúa Gates (2025) en su artículo celebrando la efeméride, ambos conocían BASIC porque fue el primer lenguaje de programación que aprendieron, así que desarrollaron el intérprete de BASIC para el Altair 8800 que licenciaron a MITS en 1975.

La decisión, desde el punto de vista empresarial, no parecía la más acertada en ese momento ya que, tal como se ha mostrado en la Figura 2, si se ampliaba el intérprete de BASIC a otros ordenadores diferentes al Altair 8800, implicaría volver a escribirlo para cada uno de ellos, lo que supondría una inversión de tiempo y dinero importante. Además, la decisión de optar por un lenguaje de alto nivel fácil de aprender como BASIC (Kurtz, 1978) era una apuesta por la democratización de la informática, ya que abría las puertas a que otras compañías, aficionados y aficionadas desarrollaran software para el Altair 8800, lo que aumentaría la competencia en el incipiente sector. También a la educación, ya que esta democratización de la informática permitía introducir procesos de enseñanza-aprendizaje del software informático en términos genéricos, lo que se conoce como el proceso de alfabetización informática (López *et al.*, 2021), y que si se concretan a la enseñanza-aprendizaje de la programación informática nos encontramos ante el desarrollo del pensamiento computacional (Segura *et al.*, 2019).

Sin embargo, si analizamos la presencia de BASIC en los ordenadores que surgieron tras el Altair 8800 durante la siguiente década, se puede observar que la decisión de apostar por un intérprete de BASIC generalista, en lugar de desarrollar programas específicos, supuso un acierto desde el punto de vista de penetración en el mercado de los ordenadores domésticos de 8 bits: todos los ordenadores de la Tabla 1, entre los que se pueden identificar los más populares entre 1975 y 1985, contaban con un intérprete de BASIC. En el 44,83 % de los mismos dicho intérprete era propiedad de Microsoft, y un 84,62 % de los intérpretes propiedad de Microsoft estaban almacenados en la Read Only Memory (ROM), la memoria no volátil del ordenador.

Hay diversas razones para la amplia presencia de este lenguaje tras el Altair 8800, entre las que encontramos las de índole práctico, las de estandarización y la presión de la competencia. A nivel práctico, BASIC no solo permitía programar, sino que también ofrecía a usuarios y usuarias la posibilidad de interactuar con el ordenador en una época en la que los sistemas operativos en el ámbito de la informática doméstica no estaban ampliamente extendidos (Hansen, 2001); en este sentido, BASIC cumplía las funciones mínimas de un sistema operativo. Respecto a la estandarización, si analizamos el esquema propuesto en la

¹ Cita traducida por los autores del texto.

Figura 2, que un ordenador contara con un intérprete de BASIC permitía que la máquina pudiera ejecutar una cantidad importante de software escrito para dicho lenguaje, sin necesidad de contar con implementaciones específicas para el hardware escritas en ensamblador; aquí, la productividad de las empresas de la que hablaba Sebesta (2007), se traduciría en un amplio catálogo de software para aquellas máquinas con intérprete de BASIC. Respecto a la presión de la competencia, la presente investigación mostrará que BASIC aparecía en un amplio porcentaje de anuncios publicitarios de ordenadores, lo que pudo suponer un elemento de presión tácita para los nuevos fabricantes que se veían obligados a incluir un intérprete de BASIC para no quedar al margen.

Por estos motivos, BASIC se convirtió en una de las piedras angulares de la naciente informática de consumo, y la apuesta de la compañía recién fundada por Gates y Allen (Micro-soft originalmente) por lo que suponía este lenguaje a la democratización de la informática, posicionó a la compañía en un lugar clave en el nuevo sector: el 44,83 % de presencia de Microsoft BASIC aventajó ampliamente a otros fabricantes de intérpretes de BASIC, como Sinclair (10,34 %), BBC (6,90 %) o Sharp (6,90 %).

Tabla 1.
Ordenadores de la década 1975-1985 con BASIC

Ordenador	Año	Desarrollo/Licencia	En ROM
Altair 8800	1975	Microsoft	
Apple I	1976	Steve Wozniak	
Commodore PET	1977	Microsoft-Commodore	X
Tandy TRS-80	1977	Microsoft	X
Apple II	1977	Apple-Microsoft	X
Sharp MZ-80k	1978	Sharp	
IBM 5110	1978	IBM	
NEC PC-8001	1979	Microsoft	X
Atari 400/800	1979	Atari	
Commodore VIC-20	1980	Microsoft-Commodore	X
Hewlett-Packard HP-85	1980	HP	
Sinclair ZX80	1980	Sinclair	
IBP PC (IBM 5150)	1981	IBM-Microsoft	X
BBC Micro	1981	BBC	
Sinclair ZX81	1981	Sinclair	
Osborne 1	1981	Microsoft	
Commodore 64	1982	Microsoft-Commodore	X
Dragon 32/64	1982	Microsoft	X
Sharp X1	1983	Sharp	
Coleco Adam	1983	Lazer Microsystems	
MSX	1983	Microsoft	X
Amstrad CPC 464	1984	Amstrad	
Sinclair QL	1984	Sinclair	X
Apple Macintosh	1984	Apple	
Tatung Einstein	1984	Crystal Research	
Acorn Electron	1984	BBC	
Commodore Plus/4	1984	Commodore	X
Toshiba HX-10	1984	Microsoft	X
Sharp MZ-700	1985	Microsoft	X

Fuente: elaboración propia.

Sin embargo, que el lenguaje de programación de Microsoft tuviera una presencia tan amplia en la primera generación de ordenadores domésticos no fue suficiente para asegurar su expansión. A pesar de que era sencillo de programar, usuarios y usuarias necesitaban formación. Esta, salvo contadas excepciones y circunscritas al marco de la educación superior como la de Dartmouth College expuesta previamente, o la de 1975 en la John F. Kennedy High School en Somers (Lett y Gregor, 1976), no vino de la mano del aprendizaje formal ni se incorporó al currículum escolar hasta años más tarde, sino que diversas iniciativas en forma de revistas, clubs y libros vinieron a cubrir esta necesidad formativa y exploraron diversas vías para enseñar a programar estos nuevos dispositivos en un marco de aprendizaje informal.

3.La enseñanza de la programación en el marco informal: fundamentación metodológica

En el contexto del aprendizaje a lo largo de toda la vida, entendido como un proceso continuo que va más allá de los límites temporales de la educación formal (Collins, 2009) y que establece una visión humanista de la educación como un proceso que está en constante evolución a lo largo de la vida (Faure, 1972), los aprendizajes no formales e informales cobran una especial relevancia. En este sentido, Siemens (2005) afirma que el aprendizaje formal ya no cubre la mayor parte de los procesos de adquisición de conocimiento en la vida de las personas. Mientras que el aprendizaje informal se caracteriza por su falta de sistematización, y el formal por estar circunscrito a los sistemas educativos oficiales, el aprendizaje no formal se define por su sistematización al margen de los entornos educativos formales (Autor, 2022).

Desde esta perspectiva, prácticas como el aprendizaje autónomo mediante revistas técnicas, los clubs de aficionados o la lectura de libros y manuales especializados pueden considerarse formas de aprendizaje informal. Precisamente por esto, la unidad de análisis de la presente investigación han sido las revistas en las que se publicaron contenidos relacionados con BASIC y que facilitaron este tipo de procesos formativos alternativos.

Para analizar las formas mediante las cuales se articuló la enseñanza y difusión de BASIC en estos contextos informales a partir de 1975, se ha empleado un análisis de contenido de tipo mixto, combinando técnicas descriptivas de tipo cuantitativo con otras interpretativas de tipo cualitativo. Esta complementariedad metodológica permite evitar la fragmentación epistemológica del método científico (Martínez, 2005), sin que ello implique ningún conflicto entre los dos enfoques (Blanco y Pirela, 2016).

El análisis cuantitativo se ha centrado en contabilizar todos los contenidos relacionados con BASIC en las revistas de la muestra (artículos, editoriales, publicidad de ordenadores, reseñas o anuncios de libros, anuncios de clubs y menciones en cartas de lectores/as). El análisis cualitativo, por su parte, ha consistido en una revisión detallada de todos los contenidos en los que aparecía el término BASIC, con especial atención, en el caso de los artículos, a su objetivo formativo, la estrategia pedagógica utilizada, el nivel de dificultad, los recursos visuales utilizados y el público al que se dirigía.

La fuente escogida ha sido la revista estadounidense Byte, pionera en el ámbito de la divulgación informática doméstica, cuyo primer número apareció en septiembre de 1975.

Originalmente publicada por Green Publishing, aunque en el número de febrero de 1976 la editorial cambió su nombre a Byte Publications (Ranade y Nash, 1994). Mantuvo una periodicidad mensual hasta que, en 1998, ya propiedad de McGraw Hill (desde 1979), fue vendida a CPM Media y dejó de publicarse. Esta fuente resulta especialmente pertinente, ya que su primer número comienza temporalmente con la aparición de BASIC de Microsoft para el Altair 8800, y refleja a lo largo de sus primeros números la expansión y creciente influencia de este lenguaje de programación.

Para la obtención del corpus de análisis se seleccionó una muestra de los primeros 16 números de la revista Byte, publicados entre septiembre de 1975 y diciembre de 1976. El objetivo ha sido analizar, desde un enfoque longitudinal acotado y temático (McMillan y Schumacher, 2005), la evolución del tratamiento que tuvo el lenguaje BASIC durante los inicios de la informática personal. La selección se ha realizado mediante un criterio temático intencionado, incluyendo solo aquellos contenidos de cada número en los que aparece una mención explícita al término BASIC, en cualquiera de sus tipologías (artículos, editoriales, publicidad de ordenadores, reseñas o anuncios de libros, anuncios de clubes y menciones en cartas de lectores/as). Todos los números analizados incluyeron al menos un contenido relacionado con BASIC, por lo que ninguno fue excluido de la muestra.

Este enfoque ha dado lugar a una muestra compuesta por 231 contenidos relacionados con BASIC, que han aparecido publicados en los primeros 16 números de Byte.

4. Resultados

4.1. Evolución y tipología de los contenidos

A nivel cuantitativo, los primeros números analizados contaban con una periodicidad mensual y una extensión inicial de 96 páginas (números 1-9 a excepción del 4, un especial de Navidad de 112 páginas), ampliándose a 112 páginas (números 10-11), posteriormente a 128 (12-13) y finalmente a 144 (14-16).

La tipología de contenido con mayor frecuencia de aparición fue la publicidad de ordenadores ($n=124$; 53,91 %; $M=7,75$; $SD=3,08$), seguido de reseñas o anuncios de libros ($n=41$; 17,83 %; $M=2,56$; $SD=2,92$), los artículos ($n=24$; 10,39 %; $M=1,5$; $SD=1,03$), las menciones en cartas de lectores/as ($n=20$; 8,70 %; $M=1,25$; $SD=1,13$), los anuncios de clubes ($n=15$; 6,52 %; $M=1$; $SD=0,76$) y las editoriales ($n=7$; 3,04 %; $M=0,44$; $SD=0,51$). En la Figura 4 se ha representado la evolución de estas tipologías de contenidos a lo largo de los 16 primeros números.

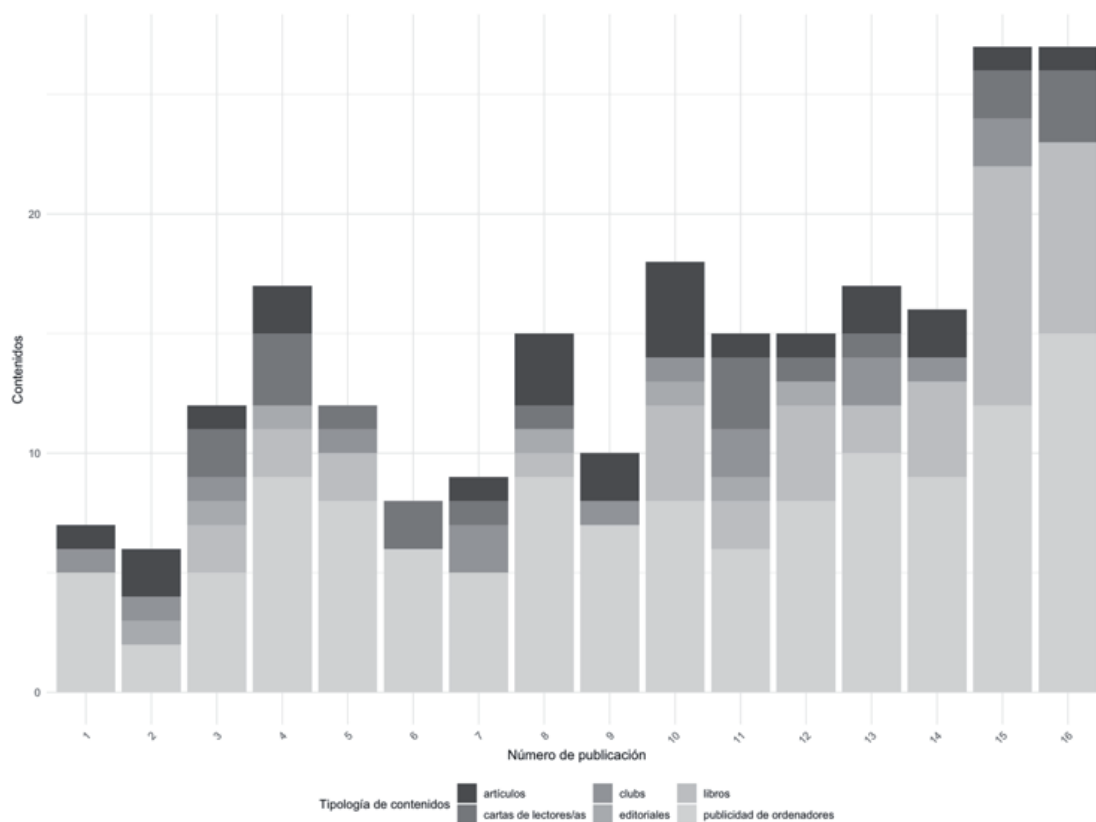


Figura 4. Evolución temporal de las tipologías de contenidos BASIC en la revista Byte. Fuente: elaboración propia.

Los dos últimos números analizados (noviembre y diciembre de 1976) fueron los que más contenidos publicaron ($n=27$ en ambos números), con un incremento del 68,75 % respecto al número de octubre de 1976. Si analizamos las tipologías, dicho incremento se debe a una eclosión de las reseñas o anuncios de libros para el aprendizaje de BASIC. De hecho, en el número de noviembre de 1976 se reseñaron o anunciaron 10 libros y en el de diciembre fueron 8, lo que contrasta con la cifra máxima de números anteriores que se situaba en 4.

Destaca la tipología de la publicidad de ordenadores: la cantidad menor de anuncios publicitarios la encontramos en el número de octubre de 1975 ($n=2$), mientras que el mayor se sitúa en el número de diciembre de 1976 ($n=15$) coincidiendo con la campaña navideña, con una amplia diferencia respecto al resto de tipologías ($M=7,75$; $SD=3,09$).

En lo que respecta a artículos, la cantidad en los que aparece el término BASIC (bien porque son el eje del mismo donde se enseñaba a realizar un programa concreto, o bien porque simplemente aparece alguna referencia al mismo) es escasa, manteniéndose siempre por debajo de 4 o menos ($M=1,5$; $SD=1,03$). Si comparamos este dato con las iniciativas del público, como por ejemplo la cantidad anuncios de clubs que realizaban los lectores y las lectoras, apenas hay diferencia entre ambas (24 artículos producidos por Byte frente a 15 anuncios de clubs). La diferencia es aún menor si nos centramos en los contenidos de dichos artículos, tal como se expondrá a continuación en el análisis cualitativo.

4.2. La búsqueda de los objetivos: ¿para qué programar con BASIC?

El hecho de que un ordenador, como máquina programable, pudiera servir para todo en función de los programas que se escribieran trajo una problemática: ¿qué se podía hacer con un ordenador? Hacerse esta pregunta era hacerse la pregunta de qué se podía hacer con BASIC, porque al fin y al cabo era un lenguaje con el que hacer programas. Así pues, se identifica durante los primeros números una búsqueda de objetivos para estas nuevas máquinas, que sería paralelamente una búsqueda de objetivos para BASIC. El análisis temático de los 24 artículos sobre BASIC publicados en Byte en sus primeros 16 números muestran que un 45,83 % (n=11) eran artículos de divulgación que trataban de responder a estas preguntas.

Los artículos de divulgación introducían a los lectores y lectoras en el mundo de la informática, primero desde una perspectiva más general guiándoles en la compra de un ordenador (Chamberlin, 1975), reflexiones sobre la calidad de los sistemas más económicos (Helmerts, 1975a) o qué variables se debía de tener en cuenta a la hora de escoger un equipo (Suding, 1976). Posteriormente se publicaron otros más específicos sobre dispositivos de hardware concretos como las unidades de almacenamiento de casete (Manly, 1976). En todos ellos estaba presente en mayor o menor medida el lenguaje de programación BASIC dando por sentado que debía ser un elemento estructural en el sistema, aunque no se diera ningún argumento. Por ejemplo, Chamberlin (1975) indicó que «one performance area of interest to hobbyists is the suitability of a machine for running a BASIC system» (p.13), pero sin dar una finalidad explícita al hecho de poder contar con un sistema BASIC. Helmerts (1975a), al indicar que «your library of programs is oriented to the larger systems which implement your BASIC interpreter» (p.90) sí que avanza este punto relacionándolo con la amplia biblioteca de programas a los que se podía acceder, pero se continúa sin dotar de un objetivo claro a los ordenadores y, por extensión, a BASIC. Incluso los mensajes que se dirigían a padres y madres también se centraban en reforzar la idea de que programar BASIC era una actividad muy motivadora, pero sin plantear unos objetivos claros. Gardner (1975) escribió que "Help them write their first BASIC program... and they're likely to be hooked for life!" (p.60). Esta línea argumentativa se mantuvo hasta que se fue perfilando la utilidad de un ordenador poco a poco. Gardner (1975) ya propuso tres posibilidades concretas: contabilidad, recetas y videojuegos.

La publicidad siguió un camino similar, siendo al principio genérica y centrándose en los aspectos técnicos de los ordenadores, pero sin dejar entrever una utilidad clara ni evidente. En la comparativa propuesta entre el número de diciembre de 1975 y el correspondiente de 1976 se puede observar que en el de 1975 se detallan aspectos técnicos del hardware, mientras que la única referencia a los programas es que su desarrollo estará supeditado a la demanda de los consumidores. En el segundo también aparecen detalles técnicos del sistema más breves, pero ahora se hace explícita la cantidad ilimitada de aplicaciones, que incluyen gráficos y computación, contabilidad, educación, tutorización y videojuegos (Figura 5).

Lowest Price in the World!

In January of 1975, MITS stunned the computer world with the announcement of the Altair 8800 Computer that sells for \$439 in kit form.

Today MITS is announcing the Altair 680. The Altair 680 built around the revolutionary new 6800 microprocessor chip, is the lowest priced complete computer on the market. Until December 31, 1975, this computer will be sold in kit form for the amazing introductory price of \$293 (A savings of \$522).

The Altair 680 comes with power supply, front panel control board, and CPU board enclosed in an 11" wide x 11" deep x 4 11/16" case. In addition to the 6800 processor, the CPU board contains the following:

1. 1024 words of memory (RAM) 2102 type 1024 x 1 bit chips).
2. Built in interface that can be configured for RS232C or 20 mA Teletype loop or 60 mA Teletype.
3. Provisions for 1024 words of ROM or PROM.

The Altair 680 can be programmed from the front panel switches or it can be connected to a computer terminal (RS232C) or a Teletype such as an ASR32 or surplus line level (Baudot) Teletype (under \$100). The Altair 680 can be utilized for many home, commercial or industrial applications or it can be used as a development system for Altair 680 CPU boards. With a cycle time of 4 microseconds, 16-bit addressing, and the capability of directly addressing 65,000 words of memory and a virtually unlimited number of I/O devices, the Altair 680 is a very versatile computer!

Altair 680 Software

Software for the Altair 680 includes a monitor on PROM, assembler, debugger, and editor. This software is available to Altair 680 owners at a nominal cost.

Future software development will be influenced by customer demand and may include BASIC on ROM. MITS will sponsor lucrative software contests to encourage the rapid growth of the Altair 680 software library. Programs in this library will be made available to all Altair 680 owners at the cost of printing and mailing.

Contact factory for updated information and prices.

Altair Users Group

All Altair 680 purchasers will receive a free one year membership to the Altair Users Group. This group is the largest of its kind in the world and includes thousands of Altair 8800 and 680 users. Members of the Altair Users Group are kept abreast of Altair developments through the monthly publication, *Computer Notes*.

Altair 680 Documentation

The Altair 680 kit comes with complete documentation including assembly manual, assembly hints manual, operation manual, and theory manual. Assembled units come with operation and theory manuals. Turnkey model and CPU boards also include documentation.

NOTE: Altair 680 manuals can be purchased separately. See back page of this catalog for prices.

Delivery

Personal checks take 2-3 weeks to process while money orders and credit card purchases can be processed in 1-3 days. Delivery should be 30-60 days but this can vary according to order backlog. All orders are handled on a first come, first served basis.



Altair 680 Prices

Altair 680 complete computer kit ... \$293 (\$545 after December 31, 1975)

Altair 680 assembled and tested	\$420
Altair 680 turnkey model (complete Altair 680 except front panel control board) Kit Only	\$240 (\$290 after December 31, 1975)
Altair 680 CPU board (including pc board, 6800 microprocessor chip, 1024 word memory, 3 way interface and all remaining components except power supply)	\$180 (\$195 after December 31, 1975)
Altair 680 CPU board assembled and tested	\$275
Option I/O socket kit (required when interfacing 680 to external devices)	\$ 29
Option cooling fan (required when expanding 680 internally)	\$ 16 (\$22 after December 31, 1975)
Option cooling fan installed	\$ 26
PROM kit (256 x 8 bit ultraviolet, erasable 1702 devices)	\$ 42

MITS
"Creative Electronics"

The Intecolor 8001 Christmas Kit.

It's an intelligent terminal for \$1395.* Or your personal computer for an additional \$1295.

THE CHOICE IS YOURS.

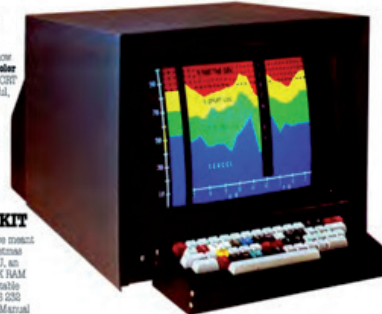
For only \$1395 you can now order the complete Intecolor 8001 8-Color, Intelligent CRT Terminal. RS that's colorful, and easy to operate. You can also invest another \$1295 to convert the Intecolor 8001 Terminal into your very own personal computer. Here's what it's all about.

THE \$1395 CHRISTMAS KIT

When we add complete we mean complete. The 8001 Christmas Kit includes the 8000 CPU, an 8-Color CRT Terminal, 4K RAM Memory, Keyboard, Selectable Read Rate up to 9600, RS 232C I/O, and an Instruction Manual which explains every assembly step. All in a single package. It's a complete stand-alone system which features our unique Intecolor 8001 VIDEO SYNCHRONIZATION SYSTEM for minimum set-up time and exceptional stability. Three to five minutes is all it takes for outstanding color registration.

THE \$1295 PERSONAL COMPUTER ADDITION

Here's what you get to convert your Intecolor 8001 Terminal into your own personal computer.



BONUS: Additional 8K RAM Work Space, Lower/Disable Background Color, Lower Case ASCII Characters, Full, 49 Line X 80 Characters/Line, 2X Character Height, and a Graphics Mode with 160X192 Elements. The applications are unlimited. Graphics and computations, check book balancing, education instructions and tutoring and a unique variety of computer games. You'll have your own personal micro computer at your fingertips. For the most simple and complex tasks.

WRITE OR CALL US TODAY

We know you're going to have some questions and we've got the answers. So contact your nearest Intecolor 8001 distributor or give us a call. We think we've come up with a great system with a price that can't be beat. Make us prove it to you.

ISC
Intelligent Systems Corp.
6045 Peachtree Corners Blvd.
Norcross, Georgia 30071
Telephone (404) 449-0801

*Domestic USA Price

Figura 5. Comparativa anuncios de ordenadores. Fuente: Byte, diciembre 1975 y diciembre 1976.

4.3. Creación de una necesidad: ¿cómo aprender BASIC?

En paralelo a los artículos analizados previamente, aparecieron otros cuyo objetivo era el desarrollo de programas relacionados con BASIC. Supusieron un 54,17 % de todos los artículos que mencionaban el término BASIC (n=13). Así pues, con unos primeros objetivos que dotaban de utilidad a los ordenadores y, por extensión, a los programas realizados con BASIC, se comenzó a enseñar en un contexto informal cómo se programaba en BASIC.

Byte realizó una progresión en cuanto a dificultad, comenzando con artículos introductorios a la programación (Helmert, 1975b), donde se impuso también la necesidad de conocer BASIC, escribiendo que "for the readers who are only just beginning to learn the programming of computers [...] a BASIC interpreter would tend to give the impression that programming is fundamentally simple. It is!" (p.34). En estos textos introductorios se trasladaba la idea de comenzar el aprendizaje con BASIC, o bien era la mejor opción para introducirse en la programación, o bien se debía de conocer el lenguaje, aunque se optara por otras soluciones.

Posteriormente a estos textos introductorios, el resto de los artículos se centraban en la realización de proyectos concretos, como por ejemplo un programa de contabilidad (Lehman, 1976), otro para la representación de biorritmos (Fox y Fox, 1976) y gráficos (Lereth, 1976) o incluso un artículo para realizar un videojuego basado en la serie de ciencia ficción Star Trek (Herd, 1976). Estos artículos se alinearon con dos de las tres propuestas de Gardner (1975), en concreto los videojuegos y la contabilidad. Todos ellos resultaban fáciles de seguir, dirigidos a personas con unos conocimientos mínimos o incluso sin conocimientos, e incluían listados con el código del programa o pseudocódigo, figuras, diagramas de flujo o imágenes del resultado del programa. Un ejemplo de la combinación de estos elementos lo encontramos en la Figura 6.

Fig. 9. The SETLIMIT routine specified in a procedure-oriented language.

```

1 SETLIMIT:
2   PROCEDURE (IROW,ICOL);
3   IF IROW < NMIN THEN NMIN = IROW;
4   IF IROW > NMAX THEN NMAX = IROW;
5   IF ICOL < NMIN THEN NMIN = ICOL;
6   IF ICOL > NMAX THEN NMAX = ICOL;
7   CLOSE SETLIMIT;

```

Subroutines Referenced by SETLIMIT:
None.

Data (8 bit bytes) used by SETLIMIT at this level:

IROW = parameter passed from FACTS_OF_LIFE.
ICOL = parameter passed from FACTS_OF_LIFE.

Data (8 bit bytes) used by SETLIMIT but shared with the whole program:

NRMIN, NRMAX, NCMIN, NCMAX (see Fig. 6)

Fig. 10. The Tree of LIFE.

is a live cell, SETLIMIT is called (see Fig. 9) in order to update the active area pointers NRMAX, NRMIN, NCMAX and NCMIN.

Where Does the LIFE Application Stand?

An alternative to the pyramid structure way of viewing program designs introduced at the beginning of this article is a "tree" notation showing the hierarchy of modules in the application. The "Tree of LIFE" is shown in Fig. 10 as it exists in materials printed to date. The next installment of LIFE Line will explore the left hand branch of the tree diagram by a similar presentation of a KEYBOARD_INTERPRETER algorithm.



Figura 6. Página de artículo que combinaba diagramas de flujo, pseudocódigo y explicaciones. Fuente: Byte, octubre 1975 (p.41).

Los artículos no solo tenían una dimensión instruccional. También se han identificado otros artículos con una dimensión pedagógica, en los que se educaba sobre los aspectos éticos relacionados con el mundo de la informática. Este es el caso del artículo titulado Are You an Author? (Mooers, 1976), en el que se trasladaba la idea de que la piratería de software era dañina para la industria y para creadores y creadoras. También se compartían ideas sobre el desarrollo de software doméstico (Wilber y Fylstra, 1976),

donde se utiliza como ejemplo la creación de una versión de BASIC mantenida por la comunidad.

A parte de los artículos, cabe destacar otro tipo de contenidos que se publicaron en Byte y que pudieron ayudar a la difusión que tuvo el lenguaje de programación BASIC. Son los anuncios que realizaban lectores y lectoras sobre la creación de clubes de programación por toda la geografía de Estados Unidos, donde los miembros compartían sus conocimientos, compendaban bibliotecas de software y realizaban cursos entre otras actividades (Figura 7).



Clubs and Newsletters

Hudson Regional Computer Fair

Computer Fairs? It just had to happen — the beginning of computer fairs for high school students, analogous to science fairs. BYTE received an announcement of the Hudson Regional Computer Fair, March 27 1976 at the White Plains Public Library, White Plains NY. Computer fairs are intended to promote and encourage the imaginative use of computers by elementary, middle and high school students. The White Plains event is sponsored by the Library, Comput-O-Mat Systems and Wang Laboratories. Entry blanks and guidelines can be obtained at the Reference Desk of the White Plains Public Library, 100 Martine Av, White Plains NY.

A Club In Michigan?

C J Lamesfield (Box 271, Davison MI 48423) would like to contact individuals interested in forming a computer club in his locale.

New Jersey's Amateur Computer Group

The ACNJ NEWS is the information organ of the Amateur Computer Group of New Jersey. The December issue reports that the November meeting of the group included demonstration of a small 6800 system by Jim Loy of Motorola. A geographical breakdown of members by county and state was also published. A summary of member systems showed that of the 129 members, 52 have microprocessor systems at home. Other activities of the Garden State organization include a software library run by Tom Kirk and amateur computer courses arranged through the Union County Technical Institute. Group purchases are also being arranged with Carl Reisinger coordinating the activity. For information on future happenings, contact Sol Libes at (201) 889-2000 (x-248, x-247 or x-282) days, and (201) 277-2063 evenings. Meetings are held the third Friday of every month — even months at Union County Technical Institute, odd months at the Middlesex County College.

New Brochure from HP-65 Users Club

The HP-65 Users Club is a non-profit volunteer group of enthusiastic programmable calculator users, designed to provide a source of information for the owners of such machines. The club has just printed a new brochure describing its objectives and activities. So far the club has centered on the HP-65; but programs and techniques concerning the HP-55, HP-25 and other programmable calculators have appeared in the newsletter 65 Notes which is published monthly. To write for the brochure (or subscribe to 65 Notes at \$12 per annum) write:

HP-65 Users Club
Richard Nelson
2541 W Camden Pl
Santa Ana CA 92704

Project SOLO

An experiment in applications of computers to education, Project SOLO is located at the University of Pittsburgh. The curious should write to Soloworks, Thomas A Dwyer, Professor of Computer Science, University of Pittsburgh, Pittsburgh PA 15260.

Seattle Club Activity

Bob Wallace, PO Box 5415, Seattle WA 98105, reports that a club is forming in the Seattle area. For information, call (206) 524-6359 11 AM to 5 PM.

Figura 7. Página de anuncios de clubes. Fuente: Byte, marzo 1976 (p.74).

4.4. BASIC se extiende a los más jóvenes: el impulso de los gráficos y los videojuegos

En el número de junio de 1976 de Byte se presenta el Cromemco TV Dazzler, un dispositivo que permitía dotar a los ordenadores personales de gráficos en blanco y negro de 128x128 pixels, o en color de 64x64 pixels. La portada del número ilustra este hito o momento clave en la historia de la informática con una simulación por ordenador del Juego de la Vida en color (Figura 8).



Figura 8. Portada mostrando el Juego de la Vida mediante el Dazzler. Fuente: Byte, junio 1976.

Después de esto aumenta significativamente el número de reseñas de libros dedicados a la programación de videojuegos mediante BASIC. En el número de noviembre de 1976 aparecen 10 reseñas o anuncios de libros para aprender BASIC, de los cuales un 70 % se centraban en el desarrollo de videojuegos. Tal como se puede observar en la Figura 9, la publicidad destacaba la posibilidad de programar juegos de palabras, de adivinar números, de patrones o de mesa entre otros. En algunos se apuntaba que era una manera mucho más atractiva de trabajar aspectos matemáticos y en otros se detallaba la estrategia pedagógica utilizada: listados de código con sus textos descriptivos y ejemplos de ejecución. Los artículos que planificó Byte no se quedaron al margen de esta nueva tendencia, ya que tres meses después, en el número de septiembre de 1976, Byte publicó un artículo sobre cómo desarrollar un videojuego basado en Star Trek (Herd, 1976).

Fun & Games with the Computer

Ted Sage. "This book is designed as a text for a one-semester course in computer programming using the BASIC language. The programs used as illustrations and exercises are games rather than mathematical algorithms, in order to make the book appealing and accessible to more students. The text is well written, with many excellent sample programs. Highly recommended." — *The Mathematics Teacher* 351 pp. \$5.95 [8B]

Game Playing With the Computer, 2nd Ed.

Donald Spencer. Over 70 games, puzzles, and mathematical recreations for the computer. Over 25 games in BASIC and FORTRAN are included complete with descriptions, flowcharts, and output. Also includes a fascinating account of the history of game-playing machines, right up to today's computer war games. Lots of "how-to" information for applying mathematical concepts to writing your own games. 320 pp. 1976 \$14.95 [8S]

ENIGMAS-1 COMPUTER GAMES IN BASIC PRICE \$8.00

.....

ENIGMAS-1 IS A BOOK OF COMPUTER GAMES TAKEN FROM MY CATALOG
THE PROGRAMS IN THIS BOOK ARE:
GONE FISHING — GO FISHING TO MAKE MONEY
CONCENTRATION — MATCH THE HIDDEN NUMBERS—TWO CAN PLAY
SLOT-MACHINE — IT'S EASY TO LOSE YOUR MONEY
CRAPS DICE GAME
WITH DICE PRINT-OUT — SHOOT CRAPS AND SEE THE DICE
TANK ATTACK — TRY TO DEFEND YOURSELF
STAR SHIP
(STAR TREK TYPE GAME) — SHOOT SOME KLINGONS
SHERLOCK HOLMES
LOGIC GAME — CHASE PROFESSOR MORI-ARTY
ALL MY PROGRAMS IN BASIC HAVE BEEN WRITTEN TO RUN IN MOST ANY BASIC COMPILER-INTERPRETER WITHOUT ANY CHANGES.
THEY WILL RUN UNDER ALTAIR 4K OR 8K BASIC AND MOST ANY OTHER BASIC COMPILER-INTERPRETER PROVIDED YOU HAVE ENOUGH MEMORY TO RUN THEM.
PRICE \$8.00 PPD. SORRY NO COD'S
B. ERICKSON
P.O. BOX 11099
CHICAGO, ILL. 60611

101 Basic Computer Games

David H. Ahl. An anthology of games and simulations—from Acey-Deucey to Yahtzee, all in the BASIC language. Contains a complete listing, sample run, plus a descriptive write-up of each game. Our most popular book! Large format, 248 pp. \$7.50 [6C]

What to Do After You Hit Return

Another collection of games and simulations—all in BASIC—including number guessing games, word games, hide-and-seek games, pattern games, board games, business and social science simulations and science fiction games. Large format. 158 pp. \$6.95 [8A]

Figura 9. Publicidad de libros para aprender a programar en BASIC. Fuente: Byte, noviembre 1976.

5. Discusión

Si, tal como se ha expuesto, los ordenadores son máquinas programables con una parte modificable (Cerrada *et al.*, 2001), el hecho de que Microsoft BASIC viniera impuesto en la memoria ROM no volátil del ordenador anulaba la posibilidad de ser eliminado.

Adicionalmente, en muchos modelos era el primer programa en cargarse cuando se arrancaba el sistema, convirtiéndose en el primer sistema operativo con el que se interactuaba con la máquina; esto implicó que el primer contacto de una cantidad importante de usuarios y usuarias con un ordenador doméstico estuvo mediatizado por Microsoft BASIC, dando a la compañía el poder de establecer las reglas iniciales con las que usuarios y usuarias interactuarían con la incipiente tecnología. En este planteamiento podemos identificar unos primeros intentos del establecimiento de nuevos espacios de rentismo para la industria *EdTech* (Saura *et al.*, 2021).

Los conceptos expuestos tras la lógica de los lenguajes de alto nivel (Sebesta, 2007) implican que el mismo código escrito en BASIC podría funcionar en diferentes ordenadores. Dicho de otra forma, un usuario o una usuaria podía cambiar de fabricante al cambiar de modelo de ordenador (hardware), pero si seguía programando en Microsoft BASIC continuaría utilizando el lenguaje del mismo proveedor (software), en este caso Microsoft. Así pues, la línea trazada por Bill Gates y Paul Allen con el cambio de paradigma hacia los lenguajes de alto nivel, aunque enmascarada tras un enfoque democratizador de la informática, no hizo sino colocar al software en el centro de la revolución informática, con una importante capacidad de influencia en las compañías que fabricaban hardware. En este sentido se ha expuesto la Tabla 1, donde el 84,62 % de los fabricantes del hardware más relevante entre 1975 y 1985 para los que se había escrito un intérprete de Microsoft BASIC, tuvieron que grabarlo en las memorias ROM que fabricaban para sus máquinas, asegurando de esta forma al consumidor que tendría acceso al lenguaje de programación al adquirir el ordenador.

El análisis cuantitativo de los orígenes de su difusión en la revista *Byte* reflejan esta ubicuidad del lenguaje de programación en la publicidad. Un dato remarcable es que más de la mitad de los contenidos de los números de las revistas analizadas en los que había alguna referencia a BASIC era publicidad de ordenadores. Es decir, el objeto que se ofrecía en dicha publicidad era un ordenador (hardware), pero se hacía referencia a la inclusión de BASIC en su ROM, o bien a las posibilidades de ejecutarlo externamente. Si nos centramos en los datos cuantitativos por número de revista, de media un 63,50 % de los contenidos publicitarios de ordenadores en cada número incluían el término BASIC. Teniendo en cuenta que los anuncios eran sobre dispositivos hardware, el dato tiene dos posibles interpretaciones: o la presencia de BASIC era un reclamo eficaz para condicionar al potencial comprador, o bien BASIC se convirtió en una exigencia tácita para los fabricantes de ordenadores.

En cualquiera de los dos casos, la exigencia de BASIC trascendió los ordenadores y llegó a aquellas personas que querían iniciarse en el mundo de la programación. Los artículos analizados seguían dos líneas diferentes: la divulgación sobre los ordenadores y el desarrollo de programas. En los primeros, tal como se ha expuesto en los análisis cualitativos, BASIC era un elemento que debía estar (Chamberlin, 1975) para acceder a una amplia biblioteca de programas (Helmerts, 1975a), pero sin dotar de utilidad clara a los ordenadores. La publicidad siguió un patrón similar, siendo al principio genérica, centrándose exclusivamente en las capacidades técnicas de las máquinas, pero no en qué es lo que se podía hacer con ellas. Esto evolucionó con el tiempo, incorporando utilidades concretas que se podía realizar con los programas y con BASIC (Figura 5). En este sentido la investigación de Navarro-Sierra *et al.* (2022) muestra una evolución similar, con la búsqueda de la construcción de la utilidad de un ordenador a través de la publicidad.

Un segundo tipo de artículos publicados en *Byte* se centraba en el desarrollo de programas en BASIC u otros lenguajes. En los casos en los que se basaban en otros lenguajes (Lehman, 1976; Lerseth, 1976), además de los listados en el lenguaje objetivo también se incluían fragmentos en pseudocódigo, que es una forma genérica de describir programas cuya traducción a otros lenguajes de alto nivel es sencilla. Aquí quedaba patente la imposibilidad de alejarse de BASIC tras la posición ubicua en la que había sido encumbrado. Un ejemplo de esta situación la encontramos en el artículo de Lehman (1976), cuando afirma que "I'll try to describe the system in enough detail so that anyone who is skilled in BASIC and knows a little about accounting could write a program to do all of the above" (p.8). También en Lerseth (1976) al afirmar que «If you have BASIC you should not have any problems implementing this routine» (p.70). Ni siquiera en los artículos que se alejaban de la dimensión instruccional para centrarse en los aspectos éticos de la informática se alejaban de la idea de necesidad de BASIC, con afirmaciones como "now appears that more copies of Altair's BASIC have been pirated than have been legally sold" (Mooers, 1976, p.16).

Sin embargo, tal como se ha mostrado en el análisis de la evolución y tipología de contenidos, el número de artículos cuyo objetivo fue la enseñanza de BASIC no fue demasiado elevado. A priori, resulta llamativa la expansión que tuvo este lenguaje de programación entre usuarios y usuarias, puesto que no se generaban tantos contenidos para el aprendizaje de BASIC en revistas como *Byte*. Helmers (1975c) arroja algo de luz sobre el tema en un artículo en la que saca a relucir las limitaciones de BASIC, pero en el que reconoce al mismo tiempo que desde sus orígenes en Dartmouth College se utilizó para enseñar conceptos de programación, y que cuenta con "innumerable tutorial books due to its widespread use in the educational field" (p.9).

Además, otras tipologías de contenidos pudieron tener una importante influencia en la expansión de BASIC, como por ejemplo los anuncios de clubes de aficionados; a pesar de ser iniciativas que se organizaban al margen de las revistas especializadas como la analizada en la presente investigación, se utilizaban estas como medio para poner en contacto a las personas. En la muestra analizada, *Byte* produjo 24 artículos relacionados con BASIC, pero los lectores y las lectoras escribieron 15 veces anunciando la creación de algún club en el que se programara con BASIC, como el ejemplo que se muestra en la Figura 7. Este tipo de actividades pudo aportar el componente social a un aprendizaje que, tal como se planteaba mediante artículos, era fundamentalmente individual. La importancia de estas comunidades de aprendizaje fue tal que el número de agosto de *Byte* le dedicó una editorial a los clubes, detallando que en los mismos las personas exponían lo que podían ofrecer a los demás y, al mismo tiempo, qué es lo que buscaban, lo que permitía poner en contacto a personas con intereses similares o complementarios que se ajustaran a sus necesidades (Helmers, 1976).

También es remarcable el incremento de las reseñas o anuncios de libros, especialmente en los últimos números analizados, con un incremento del 68,75 % respecto números anteriores. En cualquier caso, este dato debe ser tomado con cautela, puesto que en los dos últimos números se había aumentado la extensión de la revista en 16 páginas. Sin embargo, el dato es destacable también por el eje temático de los libros que se anunciaban: la programación de videojuegos.

En esta línea, la aparición del *Dazzler* pone el acento una de las posibles utilidades de los ordenadores que dio Gardner (1975): la creación de videojuegos. No resulta casual que en ese mismo número se dedica una editorial a otro videojuego, el *Tic Tac Toe*, del

que se indica que ha sido programado con BASIC. También se indica que "the Dazzer provides the hardware for an incredible variety of applications. This variety is realized through the software for games" (Byte, 1976, p.6). Con este avance, BASIC es insuflado con un abanico de nuevas posibilidades que podían atraer a un público (cliente) más joven e interesado en el mundo de los videojuegos, y que se convertiría en una impronta clave una década después (Navarro-Sierra *et al.*, 2022).

Tras el análisis realizado y los resultados en cuanto a la expansión de BASIC, se puede identificar en la estrategia empresarial de Gates y Allen algunos patrones propios de lo que hoy en día denominamos filantropcapitalismo digital, en el que actores relevantes desde el punto de vista económico imponen su visión del mundo mediante grandes donaciones económicas (Saura, 2016). Gates es uno de estos actores que en la actualidad ejerce una forma de filantropcapitalismo a través de la fundación Bill y Melina Gates.

En la década de 1970 Gates todavía no realizaba donaciones, pero la empresa que fundó junto con Allen optó por desarrollar un software que facilitaba a otras personas y empresas la tarea de producir programas informáticos. Esto iba en contra de los principios empresariales de maximizar los beneficios, ya que suponía aumentar la competencia en el sector y, por extensión, reducir los precios del software para ser competitivo.

Desde este punto de vista, la estrategia tiene algunos puntos en común con un planteamiento filantropcapitalista: desarrollar intérpretes de BASIC para diferentes tipos de ordenadores suponía apostar por la democratización de la informática en contra de mayores beneficios empresariales para Microsoft, ya que se facilitaba la entrada al sector a empresas y personas que habían aprendido a programar de una manera informal a través de revistas, clubs y libros, tal como se ha mostrado a lo largo de la investigación. Sin embargo, tras este mensaje de democratización se ocultaba un rasgo filantropcapitalista, como es el imponer una visión concreta del mundo en la que todas las personas debían tener conocimientos informáticos, y en concreto de su producto BASIC, ya que reducía las barreras de acceso a la informática (Kurtz, 1978), pero que al mismo tiempo dificultaba la búsqueda de alternativas al margen de Microsoft. Aunque excede a los objetivos de la presente investigación, simplemente apuntar que futuros movimientos de Microsoft repitieron este patrón con rasgos filantropcapitalistas. Tal es el caso de la alianza de Microsoft con IBM en 1981 para proveer al IBM-PC con su sistema operativo doméstico MS-DOS, o con Matsushita Electronics en 1983 para definir el estándar de ordenadores MSX (Lorca, 2009); ambos ejemplos resultaron clave para el nacimiento de la industria *EdTech* y la búsqueda de sus nuevos espacios rentistas, como ocurrió con el programa educativo ATENEA en España (1985-1990), pero que respondía más a intereses corporativos que pedagógicos (Trujillo, 2020). En cualquier caso, ambos movimientos de Microsoft siguen la línea iniciada con BASIC de buscar una estandarización del mercado, pero tras un mensaje de democratización de la informática, amasando con ello una gran capacidad de influencia en la industria, en la gente y en la forma en la que las personas interactuarían con los ordenadores hasta el presente.

6.Conclusiones

El desarrollo del intérprete de BASIC en 1975 por parte de Bill Gates y Paul Allen, basándose en el lenguaje creado en el Dartmouth College en 1964, marcó un punto de inflexión en la historia de la transformación digital del último cuarto del siglo XX.

La amplia presencia de BASIC en gran parte de los ordenadores de 8 bits cambió la percepción que tenía la gente sobre la informática, ya que ofrecía una manera sencilla de programar un ordenador. Tal como se ha expuesto en el presente artículo, su impacto a nivel de alfabetización informática desde la perspectiva del aprendizaje informal fue angular, ya que revistas como *Byte* publicaban contenidos en forma de artículos, editoriales, publicidad de ordenadores, reseñas o anuncios de libros, anuncios de clubes y cartas de lectores/as, todo alrededor del aprendizaje de este lenguaje de programación, lo que aumentó enormemente su popularidad en una generación de adolescentes que estaba siendo testigo del nacimiento de la informática de consumo.

Lo que a través de estos contenidos se planteó como un salto cualitativo en la democratización de la alfabetización informática al permitir que personas y empresas sin grandes conocimientos técnicos pudiesen programar ordenadores domésticos, en realidad escondía un cambio de paradigma: por primera vez, un producto intangible en forma de software se convertía en un activo poderoso que impondría sus condiciones a la incipiente industria informática. El motivo radica en la naturaleza de este lenguaje de programación: su enfoque de alto nivel permitía programar con un mismo lenguaje e instrucciones cualquier tipo de ordenador de la generación de 8 bits que contara con un intérprete de BASIC, lo que encumbró a una influyente posición a Microsoft, la compañía recién creada por Gates y Allen.

Esta estrategia colocaría a Microsoft en una situación privilegiada para imponer sus intereses en la incipiente industria, pero siempre ofuscado tras un mensaje de democratización de la alfabetización informática. El análisis de este momento clave en la historia de la informática permite arrojar luz sobre la génesis de movimientos como el filantropismo digital, ejercidos en la actualidad por el propio Gates a través de la Fundación Bill y Melinda Gates.

7. Referencias

- Blanco, N., & Pirela, J. (2016). La complementariedad metodológica: Estrategia de integración de enfoques en la investigación social. *Espacios Públicos*, 19(45), 97-111.
- Brookshear, J. G. (1993). *Teoría de la computación: lenguajes formales, autómatas y complejidad*. Adison Wesley.
- Byte (1976). About the Cover. *Byte*, 10, 6-7.
- Cerrada Somolinos, J.A., Collado Machuca, M.E., Gómez Palomo, S.R., & Estivariz López, J.F. (2001). *Fundamentos de programación cono Modula-2*. Editorial Centro de Estudios Ramón Areces.
- Chamberlin, H. (1975). Which Microprocessor is for You?. *Byte*, 1, 10-14.
- Collins, J. (2009). Lifelong learning in the 21st century and beyond. *Radiographics*, 29(2), 613-622. <https://doi.org/10.1148/rg.292085179>
- Faure, E. (1972). *Learning to be: The world of education today and tomorrow*. UNESCO.
- Fox, J., & Fox, R. (1976). Biorhythm for Computers. *Byte*, 8, 20-23.
- Gardner, R. (1975). The Shadow, Buck Rogers, and the Home Computer. *Byte*, 2, 58-60.

- Gates, W. (18 de abril de 2025). *Celebrate 50 years of Microsoft with the company's original source code*. Gates Notes. <https://www.gatesnotes.com/home/home-page-topic/reader/microsoft-original-source-code>
- Hansen, P. B. (2001). The evolution of operating systems. In *Classic operating systems: from batch processing to distributed systems* (pp. 1-34). New York: Springer New York.
- Helmets, C. (1975a). Benchmarks, Standards, etc. *Byte*, 3, 90-92.
- Helmets, C. (1975b). LIFE Line. *Byte*, 2, 34-41.
- Helmets, C. (1975c). What This Country Needs Is a Good 8-Bit High Level Language. *Byte*, 4, 5-10.
- Helmets, C. (1976). Some Notes on Clubs. *Byte*, 12, 4.
- Herd, G. (1976). A BASIC Star Trek Trainer. *Byte*, 13, 40-42.
- Knuth, D.E. (1997). *The art of computer programming, volume 1 (3rd Ed.): fundamental algorithms*. Addison Wesley Longman Publishing Co. Inc.
- Kurtz, T. E. (1978). Basic Session. In *History of programming languages* (pp. 515-537).
- Lehman, J. (1976). A Small Business Accounting System?. *Byte*, 10, 8-12.
- Lerseth, R. (1976). A plot Is Incomplete Without Characters. *Byte*, 11, 64-72.
- Lett, C., & Gregor, M. (1976). A High School Computer System. *Byte*, 10, 28-30.
- López, J. R., Ornelas, M. L., Morales, K. F., & Sandoval, J. O. (2021). Un acercamiento conceptual entre tres tipos de alfabetización: informática, tecnológica e informacional. *Texto Livre*, 14 (1), e29513.
- Lorca, F. J. (2009). Microsoft Corporation vs. the US Court of Justice and the European Community. *Jean Monnet/Robert Schuman Paper Series*, 9 (10).
- Manly, W. (1976). Magnetic Recording for Computers. *Byte*, 7, 18-27.
- Martínez, M. (2005). El paradigma emergente: Hacia una nueva teoría de la racionalidad científica. *Trillas*.
- McCracken, D. D., Stibitz, G. R., Larrivee, J. A., & Morse, P. M. (1957). Digital Computer Programming and Mathematics and Computers. *Physics Today*, 10(9), 32-32.
- McMillan, J.H., & Schumacher, S. (2005). *Investigación educativa: una introducción conceptual*. Pearson Educacion.
- Mooers, C. (1976). Are You an Author?. *Byte*, 13, 16-18.
- Navarro-Sierra, N., Quevedo-Redondo, R., & Gómez-García, S. (2022). ¿Para qué sirve un ordenador? Su construcción utilitaria en la publicidad de las revistas especializadas en informática y videojuegos en España (1984-1991). *Espacio, Tiempo y Educación*, 9(2), 59-79. <https://dx.doi.org/10.14516/ete.523>

- Nevison, J. M. (1976). Computing in the Liberal Arts College: At Dartmouth, writing a computer program is part of becoming a liberally educated person. *Science*, 194(4263), 396-402.
- Ranade, J., & Nash, A. (1994). *The Best of BYTE. Two Decades on the Leading Edge*. McGraw Hill.
- Reynolds, J. C. (2008). Some thoughts on teaching programming and programming languages. *ACM Sigplan Notices*, 43(11), 108-110.
- Roberts, E., & Yates, W. (1975). Altair 8800 Minicomputer, Part I. *Popular Electronics*, 7(1), 33-49.
- Saura, G. (2016). Neoliberalismo como discurso: la evaluación en educación entre tecnologías políticas neoliberales y la nueva filantropía. *Educação, Sociedade & Culturas*, (47), 11-30.
- Saura, G., Díez Gutiérrez, E. J., & Rivera Vargas, P. (2021). Innovación tecnopedagógica 'Google'. Plataformas digitales, datos y formación docente. *REICE. Revista Electrónica Iberoamericana sobre Calidad, Eficacia y Cambio en Educación*, 19(4), 111-124. <https://doi.org/10.15366/reice2021.19.4.007>
- Sebesta, R. W. (2007). *Concepts of programming languages*. Addison-Wesley Longman Publishing.
- Segura, J. A., Nebot, M. Á. L., Mon, F. M. E., & Novella, M. G. V. (2019). El debate sobre el pensamiento computacional en educación. *RIED-Revista iberoamericana de educación a distancia*, 22(1). <https://doi.org/10.5944/ried.22.1.22303>
- Siemens, G. (2005). Connectivism: A learning theory for the digital age. *International Journal of Instructional Technology and Distance Learning*, 2(1).
- Suding, R. (1976). A Systems Approach to a Personal Microprocessor. *Byte*, 10, 32-34.
- The President's Science Advisory Committee (1967). *Computers in Higher Education*. The White House, Washington, D.C.
- Trujillo, C. M. (2020). El boom tecnológico en las escuelas de los años 80: una aproximación al programa ATENEA español. *Espacio, tiempo y Educación*, 7(1), 247-262. <http://dx.doi.org/10.14516/ete.249>
- Wilber, M., & Fylstra, D. (1976). Homebrewery vs the Software Priesthood. *Byte*, 14, 90-94.